REMARKS

Applicant amended independent claims 1, 17 and 22 for greater clarity. Claims 1-4, 6-8, 10-14 and 17-26 are pending. Claims 1, 17, and 22 are independent.

The examiner rejected claims 1-4, 6-8, 11-14 and 17-23 under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 5,928,358 to Takayama, in view of U.S. Patent No. 5,724,563 to Hasegawa. The examiner also rejected claim 10 under 35 U.S.C. §103(a) as being unpatentable over Takayama in view of Hasegawa, and further in view of U.S. Patent No. 5,274,770 to Khim Yeoh et al. The examiner further rejected claims 24-26 under 35 U.S.C. §103(a) as being unpatentable over Takayama in view of Hasegawa, and further in view of U.S. Patent No. 5,923,872 to Chrysos.

The examiner stated:

> 3. As to cliams 1,9, 20, 22, taught at leasy [*sic*] :
> a) executing a branch instruction [20] in executon of an instruction stream (see the sequence of progrma in col.7, lines 32-37) with a branch based on a bit (I bit branch prediction 20b) specified in the branch instruction of a register specified in the branch instrulction being set or cleared (see col.7, lines 38-51), and including a first token [20d] that specified destination addres [*sic*] of instruction in the instruction stream that are after the branch instruction to execute and a second token that specified a branch guess operation (see the frequency of tken [*sic*] branch in the history in col.7, ines 52-63 as the guess informaiton, for xecution see instruciton [*sic*] execution in col.8, lines 49-65). (Office Action, page 2, paragraph 3)

Applicant disagrees.

Applicant's independent claim 1 recites "executing a branch instruction in execution of an instruction stream with a branch based on a bit of a register being set or cleared, the bit and the register being specified in the branch instruction." Thus, applicant's claim is directed to executing a branch instruction that identifies a particular bit location, of a particular register, whose content (i.e. "1" or "0") will determine if the branching operation is performed.

In contrast, Takayama describes an apparatus to facilitate branching prediction functionality. Particularly, as describes in the Abstract:

> **A branch instruction includes a set of branch prediction information 13b and a set of branch history information 13c. The set of branch prediction information 13b is made up of 1 bit which predicts whether a branch will be performed during the next execution of the instruction. The set of branch history information 13c is made up of 2 bits showing a frequency, with which the branch has been taken, is "very high", "high", "low" or "very**

> **low". An instruction fetching unit 12 prefetches an instruction from a cache memory 11a in accordance with the set of branch prediction information 13b. After an instruction executing unit 15 completes an execution of the branch instruction, a branch history information generating unit 16 generates a new set of branch history information and a branch prediction information generating unit 17 generates a new set of branch prediction information, in accordance with the execution result and the preceding branch history information 13c. A branch instruction updating unit 18 overwrites the generated set of branch history information and the generated set of branch prediction information on the corresponding branch instruction stored in the main memory 11a.**

Takayama further explains that:

> **FIG. 3 shows the branch instruction format of the information processing apparatus 100. A branch instruction 20 is composed of a 13-bit operation code 20a, 1-bit branch prediction information 20b, 2-bit branch history information 20c and a 16-bit branch destination address 20d. The operation code 20a shows an operation code and a branch condition which identify the present instruction.**
>
> **The branch prediction information 20b predicts whether the branch is taken or not taken when the present branch instruction is next executed. The relation between the branch prediction information 20b and the prediction is as follows: when the branch prediction information 20b is "0", a branch is predicted "not taken"; and when the branch prediction information 20b is "1", a branch is predicted "taken". (Col. 7, lines 38-44)**

Takayama's branch prediction information merely indicates to the apparatus what is the expected *a priori* result of the branch condition that is still to be evaluated. If the *a priori* prediction is that the branch will be taken (i.e., the condition to be evaluated is one that will cause the branching to be performed), then the next instruction from the instruction stream that Takayama's apparatus fetches is the instruction at the branched-out address. The prediction information provides a prediction of whether an instruction steam will branch or not. In contrast, claim 1 is directed to executing a type of instruction, i.e., a branch instruction during execution of instruction stream. Execution of the branch instruction predicates the branch based on a bit being set or cleared, the bit and the register being specified in the branch instruction.

Thus, Takayama's branch prediction operation is not executing of a branch instruction, but merely a precursor to the branch operation. Furthermore, contrary to the examiner's contention, the branch prediction bit does not specify a particular bit of a particular register, whose content is to be evaluated. Rather, the branch prediction bit just indicates the expected result of the to-be-evaluated branching condition. That branch prediction bit does not itself

participate in the execution of the branching operation itself. Accordingly, Takayama neither disclosed nor suggests at least the feature of "executing a branch instruction in execution of an instruction stream with a branch based on a bit of a register being set or cleared, the bit and the register being specified in the branch instruction," as required by applicant's independent claim 1.

As explained in applicant previous Amendment in Reply, Hasegawa discloses a pipeline processor that can execute predictive branch instructions (Abstract). Hasegawa further describes that the format of its predictive branch instructions includes a region 21 that stores an opcode, a region 22 for specifying a branch target address, and a region 23 for storing the number of at least one instruction which is to be executed in succession after the predictive branch instruction and before the control flow is changed (FIG. 2 and col. 6, lines 1-6). Hasegawa's predictive branch instruction, however, does not include a region for specifying a bit and/or specifying a register.

Furthermore, Hasegawa's predictive branching is performed based on the execution result 109 produced by executing an instruction on execution section 11 of Hasegawa's pipeline processor (FIG. 9, col. 11). Specifically, the execution of an instruction on execution section 11 (the executed instruction is not the predictive branch instruction) causes a number of flags on the condition code 61 of the judging section 13 to be set in accordance with the result outcome produced by the execution of the instruction (FIG. 9 and col. 11, lines 20-32). These flags include the Zero flag Z, the negative flag N, the Carryover flag C, and the overflow flag V.

Hasegawa further explains that after the execution result 109 is produced, the opcode in the region 21 of the predictive branch instruction is input from the instruction decoding section 3 to the branch condition judging section 62 (col. 11, lines 33-35). For example, if the opcode received from the predictive branching instruction is "100", branching will occur if the Z flag has been set as a result of the execution of a preceding instruction (see Table 1 at col. 11). Thus, Hasegawa's branching decisions are based on the value of the Z, N, C and V flags of the condition code 61 unit, and not on the value of a particular bit of a particular register as specified by the branching instruction.

Accordingly, Hasegawa does not disclose or suggest at least the feature of "executing a branch instruction in execution of an instruction stream with a branch based on a bit of a register

being set or cleared, the bit and the register being specified in the branch instruction," as required by applicant's independent claim 1.

Therefore, since neither Takayama nor Hasegawa discloses or suggests, alone or in combination, at least the feature of "executing a branch instruction in execution of an instruction stream with a branch based on a bit of a register being set or cleared, the bit and the register being specified in the branch instruction," applicant's independent claim 1 is patentable over the cited art.

Claims 2-4, 6-8, 14 and 24 depend from independent claim 1 and are therefore patentable for at least the same reasons as independent claim 1.

Independent claim 22 recites "execute a branch instruction in execution of an instruction stream in the processor, with a branch operation based on a bit of a register being set or cleared, the bit and the register being specified in the branch instruction." For reasons similar to those provided with respect to applicant's independent claim 1, at least this feature is not disclosed by the cited art. Accordingly, applicant's independent claim 22 is patentable over the cited art.

Claims 23 and 26 depend from applicant's independent claim 22 and are therefore patentable for at least the same reasons as independent claim 22.

With respect to applicant's independent claim 17, the examiner stated:

> 14. As to claim 17, Takayama taught executing a branch instruction that causes a branch operation in an instruction stream based on any specified value being true or false (see the execution unit judgi [*sic*] the condition of the branch instrcuin [*sic*] in col.13, ines 29-39). (Office Action, Page 5)

Applicant's respectfully disagrees with the examiner's contention.

With respect to the execution stage of its hardware, Takayama describes:

> **During the first half of the clock cycle 3 (3a), the instruction executing unit 15 judges the condition of the branch instruction in accordance with the control signal from the instruction decoding unit 14. After this, the instruction executing unit 15 informs the branch history information generating unit 16 and the branch prediction information generating unit 17 of the execution result (whether the branch was taken or not taken), as well as informing the instruction decoding unit 14 and the address control unit 8 of the prediction result (whether the prediction was correct or incorrect). (Col. 13, lines 29-39)**

Takayama further explains that:

The instruction decoding unit 14 is achieved by a ROM storing a microprogram or the like, and decodes the operation code 13a of an instruction stored in the instruction register 13 before outputting the decoding result as a control signal 141 to the instruction executing unit 15. When a branch instruction is subject to the decoding process, the instruction decoding unit 14 informs the instruction executing unit 15 and the address control unit 8 of the branch prediction information 13b of the branch instruction via a signal line 142. (Col. 8, lines 39-48)

Thus, one control signal provided by decoding unit 14 indicates what operation, corresponding to the decoded instruction, is to be performed. When the decoded instruction is a branch instruction, the decoding unit 14 also provides the branch prediction signal that indicates the *a priori* prediction of the branch condition evaluation outcome. But at no point does Takayama disclose that the branch operation itself, which is to be performed by the executing stage, is based on the value of a particular bit of a particular register specified by the instruction that was decoded by decoding unit 14.

Accordingly, for the foregoing reasons, and reasons similar to those provided with respect to independent claim 1, the prior art does not disclose or suggest at least the feature of "executing a branch instruction that causes a branch operation in an instruction stream based on a bit of a register being set or cleared, the bit and the register being specified in the branch instruction," as required by applicant's independent claim 17. Applicant's independent claim 17 is thus patentable over the prior art.

Claims 18-21 and 25 depend from independent claim 17 and are therefore patentable for at least the same reasons as independent claim 17.

It is believed that all the rejections and/or objections raised by the examiner have been addressed.

In view of the foregoing, applicant respectfully submits that the application is in condition for allowance and such action is respectfully requested at the examiner's earliest convenience.

All of the dependent claims are patentable for at least the reasons for which the claims on which they depend are patentable.

Canceled claims, if any, have been canceled without prejudice or disclaimer.

Any circumstance in which the applicant has (a) addressed certain comments of the examiner does not mean that the applicant concedes other comments of the examiner, (b) made

arguments for the patentability of some claims does not mean that there are not other good reasons for patentability of those claims and other claims, or (c) amended or canceled a claim does not mean that the applicant concedes any of the examiner's positions with respect to that claim or other claims.

No fee is believed due. Please apply any other charge or credit to deposit account 06-1050, referencing attorney docket 10559-311US1.

Respectfully submitted,

Date: Aug. 2, 2006

Ido Rabinovitch
Attorney for Intel Corporation
Reg. No. L0080

PTO Customer No. 20985
Fish & Richardson P.C.
Telephone: (617) 542-5070
Facsimile: (617) 542-8906